AMENDMENTS TO THE SPECIFICATION

Please amend the paragraph beginning on page 3, line 9, as follows:

In accordance with the present invention, an API call replay tool facilitates creating and submitting API calls based upon input API call records. The API call replay tool ensures that the API calls, executed from potentially a sequence of logged API calls, replay in a meaningful manner. This is generally accomplished by translating addresses (memory references) specified by recorded API calls into addresses within the API call replay tool's memory space. More particularly, the API call replay tool includes a symbol table for mapping references within an input API call record into a memory space allocated to the API call replay tool. In a particular embodiment, such mapping occurs from a recorded address to a replay address space allocated to a thread with which the API call is associated.

Please amend the paragraph beginning on page 5, line 11, as follows:

In summary of the system disclosed herein below, the replay tool is a core component within a test system that takes as input a set of input API call records (described with reference to FIG. 3). The input API call records include both the API call as well as information sufficient to enable re-creation of the environment within which the API is to be executed when the API call is replayed. While the API call records can be logged by an API call capture mechanism during execution of an application program, the source of the API call records is not material to the API replay tool.

Please amend the paragraph beginning on page 5, line 18, as follows:

The API replay tool includes a replay engine that coordinates building memory structures for presenting the API calls to the appropriate operating system component during replay. One

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

of the functions performed by the replay engine is the mapping [[the]] of locations of parameters and structures into the memory space maintained by the API replay tool. The mapping information is stored within symbol tables.

Please amend the paragraph beginning on page 6, line 4, as follows:

Thus, as can be seen from the above summary, in an embodiment of the present invention, the replay tool receives as input a sequence of API call records. The API replay tool creates a context for a thread within which the API call will be executed. References within the original API call are mapped to the memory space of the API replay tool. An API call is created and executed according to the context and mapping created by the API replay tool. Thus, there is no need for a tester to create an executable for submitting a sequence of API calls to perform a test of an API interface. Instead, the API replay tool itself is the executable and the API call records comprise data that drives the executable API replay tool.

Please amend the paragraph beginning on page 11, line 23, as follows:

A log reader 210 opens and processes the API calls stored within the input file 206. The log reader 210, in an embodiment of the invention, reads API call records from the input file 206. The API calls retrieved from the input file 206 include pertinent replay information needed to provide a proper context for the API call including: passed parameters, returned values, a time stamp, thread, . . .etc. (see, FIG. 3 described herein below). The log reader converts the input record stream into an output stream of records, representing API calls, messages, etc., that are passed by the log reader 210 to the replay engine 212. In an embodiment of the invention, the log reader 210 processes the call records on an individual basis (i.e., in isolation without considering relationships between separate API calls). However, in an alternative embodiment,

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

multiple related/unrelated API calls are retrieved by the log reader 210 from the input file 206. The log reader 210 also retrieves the stored resources in the input file and passes them to the replay engine 212. The replay engine 212 passes the resources into a Template DLL instance (e.g., the Template DLL 226) through, for example, a generalized module loading ~~mechanism)~~ mechanism prior to executing any API calls from the input file 206.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100